

---

# Hummingbird Documentation

*Release 0.5\_dev*

**Birdhouse**

**Apr 04, 2018**



---

# Contents

---

|          |                                |           |
|----------|--------------------------------|-----------|
| <b>1</b> | <b>Installation</b>            | <b>3</b>  |
| 1.1      | Using docker-compose . . . . . | 3         |
| <b>2</b> | <b>Configuration</b>           | <b>5</b>  |
| <b>3</b> | <b>Running unit tests</b>      | <b>7</b>  |
| <b>4</b> | <b>WPS Processes</b>           | <b>9</b>  |
| 4.1      | WPS Capabilities . . . . .     | 9         |
| 4.2      | SpotChecker . . . . .          | 11        |
| 4.3      | CFChecker . . . . .            | 11        |
| 4.4      | CDO . . . . .                  | 13        |
| <b>5</b> | <b>Example: Using Docker</b>   | <b>15</b> |
| <b>6</b> | <b>Sphinx AutoAPI Index</b>    | <b>17</b> |
| 6.1      | exceptions . . . . .           | 17        |
| 6.2      | processing . . . . .           | 17        |
| 6.3      | utils . . . . .                | 18        |
| 6.4      | patch . . . . .                | 18        |
| 6.5      | __init__ . . . . .             | 18        |
| 6.6      | config . . . . .               | 18        |
| 6.7      | processes . . . . .            | 18        |
| 6.8      | tests . . . . .                | 22        |
| <b>7</b> | <b>Indices and tables</b>      | <b>25</b> |
|          | <b>Python Module Index</b>     | <b>27</b> |



**Hummingbird (the bird)** *Hummingbirds are among the smallest of birds. They hover in mid-air at rapid wing flapping rates, typically around 50 times per second, but possibly as high as 200 times per second, allowing them also to fly at speeds exceeding 15 m/s (54 km/h; 34 mph), backwards or upside down. [..]. (Wikipedia).*

Hummingbird provides a [Web Processing Service \(WPS\)](#) using [PyWPS](#) for the climate science community. It has WPS processes for common tools used in climate science like [CDO](#) and [cfchecker](#).

Hummingbird is part of the [Birdhouse](#) project.

Contents:



# CHAPTER 1

---

## Installation

---

Check out code from the Hummingbird github repo and start the installation:

```
$ git clone https://github.com/bird-house/hummingbird.git
$ cd hummingbird
$ make clean install
```

For other install options run `make help` and read the documentation of the [Makefile](#).

After successful installation you need to start the services. Hummingbird is using [Anaconda](#) Python distribution system. All installed files (config etc ...) are below the Anaconda root folder which is by default in your home directory `~/anaconda`. Now, start the services:

```
$ make start    # starts supervisor services
$ make status  # shows supervisor status
```

The depolyed WPS service is by default available on <http://localhost:8092/wps?service=WPS&version=1.0.0&request=GetCapabilities>.

Check the log files for errors:

```
$ tail -f ~/birdhouse/var/log/pywps/hummingbird.log
$ tail -f ~/birdhouse/var/log/supervisor/hummingbird.log
```

## 1.1 Using docker-compose

Start hummingbird with docker-compose (docker-compose version > 1.7):

```
$ docker-compose up
```

By default the WPS is available on port 8080: <http://localhost:8080/wps?service=WPS&version=1.0.0&request=GetCapabilities>.

You can change the ports and hostname with environment variables:

```
$ HOSTNAME=hummingbird HTTP_PORT=8092 SUPERVISOR_PORT=48092 docker-compose up
```

Now the WPS is available on port 8092: <http://hummingbird:8092/wps?service=WPS&version=1.0.0&request=GetCapabilities>.



## CHAPTER 2

---

### Configuration

---

If you want to run on a different hostname or port then change the default values in `custom.cfg`:

```
$ cd hummingbird
$ vim custom.cfg
$ cat custom.cfg
[settings]
hostname = localhost
http-port = 8092
```

After any change to your `custom.cfg` you **need** to run `make update` again and restart the supervisor service:

```
$ make update    # or install
$ make restart
```



## CHAPTER 3

---

### Running unit tests

---

**Run quick tests:**

```
$ make test
```

**Run all tests (slow, online):**

```
$ make testall
```

**Check pep8:**

```
$ make pep8
```



We describe here the WPS processes available in Hummingbird.

## 4.1 WPS Capabilities

Using the default Hummingbird installation the GetCapabilities request is as follows:

<http://localhost:8092/wps?service=WPS&version=1.0.0&request=GetCapabilities>

The XML response of the WPS service is the following document:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <wps:Capabilities service="WPS" version="1.0.0" xml:lang="en-CA" xmlns:xlink="http://
   ↳ www.w3.org/1999/xlink" xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows=
   ↳ "http://www.opengis.net/ows/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   ↳ instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.
   ↳ opengis.net/wps/1.0.0/wpsGetCapabilities_response.xsd" updateSequence="1">
3     <ows:ServiceIdentification>
4         <ows:Title>Hummingbird</ows:Title>
5         <ows:Abstract>WPS processes for general tools used in the climate_
   ↳ science community like cdo</ows:Abstract>
6         <ows:Keywords>
7             <ows:Keyword>WPS</ows:Keyword>
8             <ows:Keyword>PyWPS</ows:Keyword>
9         </ows:Keywords>
10        <ows:ServiceType>WPS</ows:ServiceType>
11        <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
12        <ows:Fees>None</ows:Fees>
13        <ows:AccessConstraints>None</ows:AccessConstraints>
14    </ows:ServiceIdentification>
15    <ows:ServiceProvider>
16        <ows:ProviderName></ows:ProviderName>
17        <ows:ServiceContact>
18            <ows:ContactInfo>

```

```

19         <ows:HoursOfService>0:00-24:00</ows:HoursOfService>
20         <ows:ContactInstructions>None</
↳ows:ContactInstructions>
21     </ows:ContactInfo>
22     </ows:ServiceContact>
23 </ows:ServiceProvider>
24 <ows:OperationsMetadata>
25     <ows:Operation name="GetCapabilities">
26         <ows:DCP>
27             <ows:HTTP>
28                 <ows:Get xlink:href="http://localhost:8092/
↳wps?"/>
29                 <ows:Post xlink:href="http://localhost:8092/
↳wps"/>
30             </ows:HTTP>
31         </ows:DCP>
32     </ows:Operation>
33     <ows:Operation name="DescribeProcess">
34         <ows:DCP>
35             <ows:HTTP>
36                 <ows:Get xlink:href="http://localhost:8092/
↳wps?"/>
37                 <ows:Post xlink:href="http://localhost:8092/
↳wps"/>
38             </ows:HTTP>
39         </ows:DCP>
40     </ows:Operation>
41     <ows:Operation name="Execute">
42         <ows:DCP>
43             <ows:HTTP>
44                 <ows:Get xlink:href="http://localhost:8092/
↳wps?"/>
45                 <ows:Post xlink:href="http://localhost:8092/
↳wps"/>
46             </ows:HTTP>
47         </ows:DCP>
48     </ows:Operation>
49 </ows:OperationsMetadata>
50 <wps:ProcessOfferings>
51     <wps:Process wps:processVersion="0.1">
52         <ows:Identifier>cdo_sinfo</ows:Identifier>
53         <ows:Title>CDO sinfo</ows:Title>
54         <ows:Abstract>Apply CDO sinfo on NetCDF File.</ows:Abstract>
55         <ows:Metadata xlink:title="CDO" xlink:href="https://code.zmaw.
↳de/projects/cdo" />
56     </wps:Process>
57     <wps:Process wps:processVersion="0.1">
58         <ows:Identifier>cdo_operation</ows:Identifier>
59         <ows:Title>CDO Operation</ows:Title>
60         <ows:Abstract>Apply CDO Operation like monmax on NetCDF File.
↳</ows:Abstract>
61         <ows:Metadata xlink:title="CDO" xlink:href="https://code.zmaw.
↳de/projects/cdo" />
62     </wps:Process>
63     <wps:Process wps:processVersion="0.1">
64         <ows:Identifier>cfchecker</ows:Identifier>
65         <ows:Title>CF Checker</ows:Title>
66         <ows:Abstract>The cfchecker checks NetCDF files for
↳compliance to the CF standard.</ows:Abstract>

```

```

67         </wps:Process>
68     </wps:ProcessOfferings>
69     <wps:Languages>
70         <wps:Default>
71             <ows:Language>en-CA</ows:Language>
72         </wps:Default>
73         <wps:Supported>
74             <ows:Language>en-CA</ows:Language>
75         </wps:Supported>
76     </wps:Languages>
77     <wps:WSDL xlink:href="http://localhost:8092/wps?WSDL"/>
78 </wps:Capabilities>

```

## 4.2 SpotChecker

Spot Checker checks a single dataset (NetCDF or OpenDAP) against a variety of compliance standards. Available compliance standards are the Climate and Forecast conventions (CF) and project specific rules for CMIP6 and CORDEX.

## 4.3 CFChecker

The `cfchecker` checks NetCDF files for compliance to the Climate Forecast Conventions (CF) standard.

The process expects one or more NetCDF files which should be checked and an optional parameter for the CF version.

### 4.3.1 WPS process description

Using the default Hummingbird installation the DescribeProcess request is as follows:

`http://localhost:8092/wps?service=WPS&version=1.0.0&request=DescribeProcess&identifier=cfchecker`

The XML response of the WPS service is the following document:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <wps:ProcessDescriptions xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows=
  ↳ "http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  ↳ www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_
  ↳ response.xsd" service="WPS" version="1.0.0" xml:lang="en-CA">
3     <ProcessDescription wps:processVersion="0.1" storeSupported="true"
  ↳ statusSupported="true">
4         <ows:Identifier>cfchecker</ows:Identifier>
5         <ows:Title>CF Checker</ows:Title>
6         <ows:Abstract>The cfchecker checks NetCDF files for compliance to the CF
  ↳ standard.</ows:Abstract>
7         <DataInputs>
8             <Input minOccurs="1" maxOccurs="1000">
9                 <ows:Identifier>resource</ows:Identifier>
10                <ows:Title>NetCDF File</ows:Title>
11                <ows:Abstract>NetCDF File</ows:Abstract>
12                <ComplexData>
13                    <Default>
14                        <Format>

```

```

15         <MimeType>application/x-netcdf</MimeType>
16     </Format>
17 </Default>
18 <Supported>
19     <Format>
20         <MimeType>application/x-netcdf</MimeType>
21     </Format>
22 </Supported>
23 </ComplexData>
24 </Input>
25 <Input minOccurs="0" maxOccurs="1">
26     <ows:Identifier>cf_version</ows:Identifier>
27     <ows:Title>CF version</ows:Title>
28     <ows:Abstract>CF version to check against, use auto to auto-detect_
↔the file version.</ows:Abstract>
29     <LiteralData>
30         <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/
↔#string">string</ows:DataType>
31         <ows:AllowedValues>
32             <ows:Value>auto</ows:Value>
33             <ows:Value>1.6</ows:Value>
34             <ows:Value>1.5</ows:Value>
35             <ows:Value>1.4</ows:Value>
36             <ows:Value>1.3</ows:Value>
37             <ows:Value>1.2</ows:Value>
38             <ows:Value>1.1</ows:Value>
39         </ows:AllowedValues>
40         <DefaultValue>auto</DefaultValue>
41     </LiteralData>
42 </Input>
43 </DataInputs>
44 <ProcessOutputs>
45     <Output>
46         <ows:Identifier>output</ows:Identifier>
47         <ows:Title>CF Checker Report</ows:Title>
48         <ComplexOutput>
49             <Default>
50                 <Format>
51                     <MimeType>text/plain</MimeType>
52                 </Format>
53             </Default>
54             <Supported>
55                 <Format>
56                     <MimeType>text/plain</MimeType>
57                 </Format>
58             </Supported>
59         </ComplexOutput>
60     </Output>
61 </ProcessOutputs>
62 </ProcessDescription>
63 </wps:ProcessDescriptions>

```

The WPS Parameters are:

**resource** Is the input parameter to provide one or more URLs (`http://`, `file://`) to NetCDF files. It is a WPS `ComplexData` type with MIME-type `application/x-netcdf`.

**cf\_version** Is an optional input parameter to provide the CF version to check against. It is a WPS `LiteralData` type with a set of allowed values (1.1, 1.2, ..., auto).



*output* Is the output parameter to provide the report of the CF check as text document. It is a WPS ComplexData type with MIME-type `text/plain`.

### 4.3.2 WPS process execution

An example execution of the `cfchecker` process with public available data:

```
http://localhost:8092/wps?service=WPS&version=1.0.0&request=Execute&identifier=cfchecker&DataInputs=resource=http://www.esrl.noaa.gov/psd/thredds/fileServer/Datasets/ncep/vwnd.sfc.2015.nc&RawDataOutput=output
```

The process is called with key/value parameters, synchronously and with direct output (`RawDataOutput`).

The resulting text document of the `cfchecker` report looks like the following:

## 4.4 CDO

to be continued ...



---

## Example: Using Docker

---

If you just want to try the Hummingbird Web Processing Service you can also use the [Docker](#) image:

```
$ docker run -i -d -p 9001:9001 -p 8000:8000 -p 8080:8080 birdhouse/hummingbird
```

Open your browser and enter the url of the supervisor service:

<http://localhost:9001/>

Run a GetCapabilities WPS request:

<http://localhost:8080/wps?service=WPS&version=1.0.0&request=getcapabilities>

Run DescribeProcess WPS request for *CFChecker*:

<http://localhost:8080/wps?service=WPS&version=1.0.0&request=describeprocess&identifier=cfchecker>

Execute *CFChecker* process with public available data:

<http://localhost:8080/wps?service=WPS&version=1.0.0&request=Execute&identifier=cfchecker&DataInputs=dataset=http://www.esrl.noaa.gov/psd/thredds/fileServer/Datasets/ncep/vwnd.sfc.2015.nc&RawDataOutput=output>

Install *Birdy* WPS command line tool from Anaconda (Anaconda needs to be installed and in your PATH):

```
$ conda install -c birdhouse birdhouse-birdy
```

Use *Birdy* to access Hummingbird WPS service:

```
$ export WPS_SERVICE=http://localhost:8080/wps
$ birdy -h
$ birdy cfchecker -h
$ birdy cfchecker --dataset http://www.esrl.noaa.gov/psd/thredds/fileServer/Datasets/
↳ ncep/vwnd.sfc.2015.nc
```



This page is the top-level of your generated API documentation. Below is a list of all items that are documented here.

## 6.1 exceptions

### 6.1.1 Module Contents

**class** `ProcessError`

## 6.2 processing

### 6.2.1 Module Contents

`processing.get_cdo()`

`processing.ncdump(dataset)`

Returns the metadata of the dataset

Code taken from <https://github.com/ioos/compliance-checker-web>

`processing.cmor_tables_path()`

`processing.cmor_tables()`

`processing.cmor_dump_output(dataset, status, output, output_filename)`

`processing.cmor_checker(dataset, table="CMIP6_CV", variable=None, output_filename=None)`

`processing.hdh_cf_check(filename, version="auto")`

`processing.hdh_qa_checker(filename, project, qa_home=None)`

## 6.3 `utils`

### 6.3.1 Module Contents

`utils.make_dirs` (*path*)

`utils.fix_filename` (*filename*)

`utils.output_filename` (*filename*, *addition=None*, *extension=None*)  
build an appropriate output filename based on filename, addition and extension.

## 6.4 `patch`

### 6.4.1 Module Contents

`patch.patch_compliance_checker` ()  
Patch compliance\_checker for ESGF OpenDAP with `.dodsrc`.

## 6.5 `__init__`

### 6.5.1 Package Contents

`__init__.application` (*environ*, *start\_response*)

## 6.6 `config`

### 6.6.1 Module Contents

`config.cache_path` ()

## 6.7 `processes`

### 6.7.1 Submodules

`processes.wps_cdo_bbox`

Processes with cdo commands

#### Module Contents

**class** `processes.wps_cdo_bbox.CDOBBBox`  
This process calls `cdo sellonlatbox` on netcdf file  
`_handler` (*request*, *response*)

### `processes.wps_cdo_copy`

Processes with cdo commands

#### Module Contents

**class** `processes.wps_cdo_copy.CDOCopy`

This process calls cdo with the copy operation on datasets (netcdf, opendap)

`_handler` (*request, response*)

### `processes.wps_cdo_ensembles`

Processes with cdo ensemble operations

#### Module Contents

**class** `processes.wps_cdo_ensembles.CDOEnsembles`

`_handler` (*request, response*)

### `processes.wps_cdo_indices`

Processes with cdo commands

#### Module Contents

**class** `processes.wps_cdo_indices.CDOClimateIndices`

This process calls cdo to calculate climate indices written to a netcdf file

`_handler` (*request, response*)

### `processes.wps_cdo_inter_pywps4`

Processes with cdo commands

#### Module Contents

`processes.wps_cdo_inter_pywps4.cdo_wrap` (*tmargs*)

**class** `processes.wps_cdo_inter_pywps4.CDOinter_MPI`

`_handler` (*request, response*)

### `processes.wps_cdo_op`

Processes with cdo commands

## Module Contents

**class** `processes.wps_cdo_op.CDOOperation`  
This process calls cdo with operation on netcdf file  
`_handler` (*request, response*)

`processes.wps_cdo_sinfo`

Processes with cdo commands

## Module Contents

**class** `processes.wps_cdo_sinfo.CDOInfo`  
This process calls cdo sinfo on netcdf file  
`_handler` (*request, response*)

`processes.wps_cfchecker`

## Module Contents

`processes.wps_cfchecker.cf_check` (*nc\_file, version*)

**class** `processes.wps_cfchecker.CFChecker`  
`_handler` (*request, response*)

`processes.wps_cmor_checker`

## Module Contents

**class** `processes.wps_cmor_checker.CMORChecker`  
`_handler` (*request, response*)

`processes.wps_compliance_checker`

## Module Contents

**class** `processes.wps_compliance_checker.CChecker`  
`_handler` (*request, response*)



`processes.wps_hdh_cfchecker`

### Module Contents

`class` `processes.wps_hdh_cfchecker.HDHCFChecker`

`_handler` (*request, response*)

`processes.wps_hdh_qachecker`

### Module Contents

`class` `processes.wps_hdh_qachecker.QualityChecker`

`_handler` (*request, response*)

`processes.wps_ncdump`

### Module Contents

`class` `processes.wps_ncdump.NCDump`

`_handler` (*request, response*)

`processes.wps_ncplot`

Processes for plotting netcdf files with matplotlib/basemap

### Module Contents

`class` `processes.wps_ncplot.SimplePlot`

Plots a simple 2D map of netcdf file

`execute` ()

`processes.wps_spotchecker`

### Module Contents

`class` `processes.wps_spotchecker.SpotChecker`

`_handler` (*request, response*)

## 6.8 tests

### 6.8.1 Submodules

`tests.common`

#### Module Contents

`class tests.common.WpsTestClient`

`get (*args, **kwargs)`

`tests.common.client_for (service)`

`tests.test_utils`

#### Module Contents

`tests.test_utils.test_fix_filename ()`

`tests.test_utils.test_output_filename ()`

`tests.test_wps_caps`

#### Module Contents

`tests.test_wps_caps.test_wps_caps ()`

`tests.test_wps_cdo_ensembles`

#### Module Contents

`tests.test_wps_cdo_ensembles.test_wps_ensembles ()`

`tests.test_wps_cdo_op`

#### Module Contents

`tests.test_wps_cdo_op.test_wps_cdo_operation ()`

`tests.test_wps_cdo_sinfo`

#### Module Contents

`tests.test_wps_cdo_sinfo.test_wps_cdo_sinfo ()`

`tests.test_wps_cfchecker`

#### Module Contents

`tests.test_wps_cfchecker.test_wps_cfchecker()`

`tests.test_wps_compliance_checker`

#### Module Contents

`tests.test_wps_compliance_checker.test_wps_cchecker()`

`tests.test_wps_hdh`

#### Module Contents

`tests.test_wps_hdh.test_wps_qa_cfchecker()`

`tests.test_wps_ncdump`

#### Module Contents

`tests.test_wps_ncdump.test_wps_ncdump()`

### 6.8.2 Package Contents

```
class tests.WpsTestCase
    Base TestCase class, sets up a wps
    setUpClass()
```



## CHAPTER 7

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



—  
\_\_init\_\_, 18

### c

config, 18

### e

exceptions, 17

### p

patch, 18  
processes, 18  
processes.wps\_cdo\_bbox, 18  
processes.wps\_cdo\_copy, 19  
processes.wps\_cdo\_ensembles, 19  
processes.wps\_cdo\_indices, 19  
processes.wps\_cdo\_inter\_pywps4, 19  
processes.wps\_cdo\_op, 19  
processes.wps\_cdo\_sinfo, 20  
processes.wps\_cfchecker, 20  
processes.wps\_cmor\_checker, 20  
processes.wps\_compliance\_checker, 20  
processes.wps\_hdh\_cfchecker, 21  
processes.wps\_hdh\_qachecker, 21  
processes.wps\_ncdump, 21  
processes.wps\_ncplot, 21  
processes.wps\_spotchecker, 21  
processing, 17

### t

tests, 22  
tests.common, 22  
tests.test\_utils, 22  
tests.test\_wps\_caps, 22  
tests.test\_wps\_cdo\_ensembles, 22  
tests.test\_wps\_cdo\_op, 22  
tests.test\_wps\_cdo\_sinfo, 22  
tests.test\_wps\_cfchecker, 23  
tests.test\_wps\_compliance\_checker, 23

tests.test\_wps\_hdh, 23  
tests.test\_wps\_ncdump, 23

### u

utils, 18





## Symbols

- \_\_init\_\_ (module), 18  
 \_handler() (processes.wps\_cdo\_bbox.CDOBBBox method), 18  
 \_handler() (processes.wps\_cdo\_copy.CDOCopy method), 19  
 \_handler() (processes.wps\_cdo\_ensembles.CDOEnsembles method), 19  
 \_handler() (processes.wps\_cdo\_indices.CDOClimateIndices method), 19  
 \_handler() (processes.wps\_cdo\_inter\_pywps4.CDOinter\_MPI method), 19  
 \_handler() (processes.wps\_cdo\_op.CDOOperation method), 20  
 \_handler() (processes.wps\_cdo\_sinfo.CDOInfo method), 20  
 \_handler() (processes.wps\_cfchecker.CFChecker method), 20  
 \_handler() (processes.wps\_cmor\_checker.CMORChecker method), 20  
 \_handler() (processes.wps\_compliance\_checker.CChecker method), 20  
 \_handler() (processes.wps\_hdh\_cfchecker.HDHCFChecker method), 21  
 \_handler() (processes.wps\_hdh\_qachecker.QualityChecker method), 21  
 \_handler() (processes.wps\_ncdump.NCDump method), 21  
 \_handler() (processes.wps\_spotchecker.SpotChecker method), 21
- A**
- application() (in module \_\_init\_\_), 18
- C**
- cache\_path() (in module config), 18  
 CChecker (class in processes.wps\_compliance\_checker), 20  
 cdo\_wrap() (in module processes.wps\_cdo\_inter\_pywps4), 19  
 CDOBBBox (class in processes.wps\_cdo\_bbox), 18  
 CDOClimateIndices (class in processes.wps\_cdo\_indices), 19  
 CDOCopy (class in processes.wps\_cdo\_copy), 19  
 CDOEnsembles (class in processes.wps\_cdo\_ensembles), 19  
 CDOInfo (class in processes.wps\_cdo\_sinfo), 20  
 CDOinter\_MPI (class in processes.wps\_cdo\_inter\_pywps4), 19  
 CDOOperation (class in processes.wps\_cdo\_op), 20  
 cf\_check() (in module processes.wps\_cfchecker), 20  
 CFChecker (class in processes.wps\_cfchecker), 20  
 client\_for() (in module tests.common), 22  
 cmor\_checker() (in module processing), 17  
 cmor\_dump\_output() (in module processing), 17  
 cmor\_tables() (in module processing), 17  
 cmor\_tables\_path() (in module processing), 17  
 CMORChecker (class in processes.wps\_cmor\_checker), 20  
 config (module), 18
- E**
- exceptions (module), 17  
 execute() (processes.wps\_ncplot.SimplePlot method), 21
- F**
- fix\_filename() (in module utils), 18
- G**
- get() (tests.common.WpsTestClient method), 22  
 get\_cdo() (in module processing), 17
- H**
- hdh\_cf\_check() (in module processing), 17  
 hdh\_qa\_checker() (in module processing), 17  
 HDHCFChecker (class in processes.wps\_hdh\_cfchecker), 21

## M

make\_dirs() (in module utils), 18

## N

NCDump (class in processes.wps\_ncdump), 21

ncdump() (in module processing), 17

## O

output\_filename() (in module utils), 18

## P

patch (module), 18

patch\_compliance\_checker() (in module patch), 18

ProcessError (class in exceptions), 17

processes (module), 18

processes.wps\_cdo\_bbox (module), 18

processes.wps\_cdo\_copy (module), 19

processes.wps\_cdo\_ensembles (module), 19

processes.wps\_cdo\_indices (module), 19

processes.wps\_cdo\_inter\_pywps4 (module), 19

processes.wps\_cdo\_op (module), 19

processes.wps\_cdo\_sinfo (module), 20

processes.wps\_cfchecker (module), 20

processes.wps\_cmor\_checker (module), 20

processes.wps\_compliance\_checker (module), 20

processes.wps\_hdh\_cfchecker (module), 21

processes.wps\_hdh\_qachecker (module), 21

processes.wps\_ncdump (module), 21

processes.wps\_ncplot (module), 21

processes.wps\_spotchecker (module), 21

processing (module), 17

## Q

QualityChecker (class in processes.wps\_hdh\_qachecker),  
21

## S

setUpClass() (tests.WpsTestCase method), 23

SimplePlot (class in processes.wps\_ncplot), 21

SpotChecker (class in processes.wps\_spotchecker), 21

## T

test\_fix\_filename() (in module tests.test\_utils), 22

test\_output\_filename() (in module tests.test\_utils), 22

test\_wps\_caps() (in module tests.test\_wps\_caps), 22

test\_wps\_cchecker() (in module  
tests.test\_wps\_compliance\_checker), 23

test\_wps\_cdo\_operation() (in module  
tests.test\_wps\_cdo\_op), 22

test\_wps\_cdo\_sinfo() (in module  
tests.test\_wps\_cdo\_sinfo), 22

test\_wps\_cfchecker() (in module  
tests.test\_wps\_cfchecker), 23

test\_wps\_ensembles() (in module  
tests.test\_wps\_cdo\_ensembles), 22

test\_wps\_ncdump() (in module tests.test\_wps\_ncdump),  
23

test\_wps\_qa\_cfchecker() (in module tests.test\_wps\_hdh),  
23

tests (module), 22

tests.common (module), 22

tests.test\_utils (module), 22

tests.test\_wps\_caps (module), 22

tests.test\_wps\_cdo\_ensembles (module), 22

tests.test\_wps\_cdo\_op (module), 22

tests.test\_wps\_cdo\_sinfo (module), 22

tests.test\_wps\_cfchecker (module), 23

tests.test\_wps\_compliance\_checker (module), 23

tests.test\_wps\_hdh (module), 23

tests.test\_wps\_ncdump (module), 23

## U

utils (module), 18

## W

WpsTestCase (class in tests), 23

WpsTestClient (class in tests.common), 22